

MANAGEMENT MODEL

Alan Delgado de Oliveira*, Tiago Pascoal Filomena
and Marcelo Brutti Righi

Received May 31, 2017 / Accepted November 22, 2017

ABSTRACT. In this paper, we provide an empirical discussion of the differences among some scenario tree-generation approaches for stochastic programming. We consider the classical Monte Carlo sampling and Moment matching methods. Moreover, we test the Resampled average approximation, which is an adaptation of Monte Carlo sampling and Monte Carlo with naive allocation strategy as the benchmark. We test the empirical effects of each approach on the stability of the problem objective function and initial portfolio allocation, using a multistage stochastic chance-constrained asset-liability management (ALM) model as the application. The Moment matching and Resampled average approximation are more stable than the other two strategies.

Keywords: scenario generation, stochastic programming, multistage, ALM.

1 INTRODUCTION

Stochastic programming (SP) is a branch of optimization, in which optimal decisions are made under uncertainty (Birge & Louveaux, 2011). In general, real world problems are characterized by uncertain events modeled with random variables. We associate a collection of random events with a probability space containing a bundle of all possible events with their probabilities. From these concepts, we formalize a generic stochastic optimization problem, as denoted by Eq. (1)

$$G(x) = \max_{x \in X} \int_F f(x, \xi) dF(\xi), \quad (1)$$

where f is the objective function that is defined in terms of uncertainty, x is the decision variable defined over the feasible set $X \subset \mathbb{R}^I$, and ξ is a random variable defined by the continuous

*Corresponding author.

Escola de Administração, Universidade Federal do Rio Grande do Sul, 90010-460 Porto Alegre, Rio Grande do Sul, Brazil. E-mails: alan.delgado@ufrgs.br; tphilomena@ea.ufrgs.br; marcelo.righi@ufrgs.br

cumulative distribution function $F : \mathbb{R}^I \rightarrow [0, 1]$. Furthermore, dF represents the probability measure of the probability space F for the underlying multivariate stochastic process.

As discussed by Pflug (2001), continuous optimization problems become easier to solve if we reduce them to discrete-state multi-period optimization problems. This logical structure may be seen as a tree model with a non-anticipative decision process, in which dependence relies uniquely on this history and on the probabilistic specification (Dupačová et al., 2000; Escudero & Kamesam, 1995). Thus, the decision functions reduce to large decision vectors and, as Löhndorf (2016) points out, it may be calculated numerically by either drawing a sample from F or by approximating F with a discrete distribution \hat{F} . Using \hat{F} instead of F , we have the following maximization problem:

$$\max_{x \in X} \sum_{\xi \in \hat{F}} \hat{\phi}(\xi) f(x, \hat{\xi}) \quad (2)$$

where $\hat{\phi}(\xi)$ is the probability of the mass point in \hat{F} . If we suppose a uniform distribution for $\hat{\phi}$, the approach in Eq. (2) can be seen as sample average approximation (SAA). Stochastic approximation (SA) is another approach to obtain numerical convergence for stochastic optimization problems, which presents a random direction whose origin is usually an objective function's gradient with a step-size for each iteration (Shapiro et al., 2009).

As the scenario generation methodology becomes a key part of the stochastic optimization process, the main goal of this study is to compare the performance of different scenario sampling methods, in order to highlight which of them is more appropriated for designing a representative discrete-space model for asset-liability management (ALM) problems regarding the in-sample performance. Many efforts have been made in the scenario generation direction, for instance, by having matched state-space distribution moments (Dupačová et al., 2000; Høyland & Wallace, 2001; Høyland et al., 2003), minimizing Wassertein probability metrics (Romisch, 2003; Heitsch & Romisch, 2005; Hochreiter & Pflug, 2007), Latin hypercube sampling (McKay et al., 1979), Voronoi cell sampling (Löhndorf, 2016), and Resampled average approximation (de Oliveira et al., 2017), among others.

This study compares the empirical results of distinct approaches to generating scenarios for ALM: Random sampling and Moment matching. We also test two Monte Carlo sampling variations: Resampled average approximation and Monte Carlo, using a naive allocation strategy as a benchmark. These sampling techniques were chosen among other well-known options, as cited above. We apply sampling methods from distinct perspectives. The Monte Carlo sampling-based algorithms are based on the uniform version of strong law of large numbers. It ensures that the optimal objective value of the SAA problem defined by Eq. (2) converges to the true value of the problem, according to the increase of the number of scenarios (Glasserman, 2003). The Moment matching depends mainly on the prior knowledge of the distribution function of the marginals. Several changes in these paradigms can be found in the literature, for instance, Mak et al. (1999) and Homem-de-Mello & Bayraksan (2014b) propose variance reduction techniques in Monte Carlo sampling-based approximations to improve the scenario representation. Moreover, Kaut &

Wallace (2011) introduce copulas in the definition of moments, but they represent perspectives with essentially distinguished methodologies.

Our intent is not to exhaustively test the sampling methods for ALM, but to outline how, empirically, the method may have an impact and produce different outputs. Based mostly on the resulting values of the objective function, as suggested by Kaut & Wallace (2007), we can conclude that the classical Monte Carlo sampling and Monte Carlo with naive allocation strategy are dominated by the Moment matching and the Resampled average approximation.

The paper is organized as follows. First, we introduce this study's multistage stochastic chance-constrained ALM model in Section 2. Then, in Section 3, the scenario generation methods are described: Monte Carlo sampling (Section 3.1), the Moment matching sampling (Section 3.2), the Resampled average approximation (Section 3.3) and the benchmark Monte Carlo with naive allocation (Section 3.4). Section 4 describes the generation of the sample paths with other explanations on the data and the experiment. A comparison of the results considering the different approaches appears in Section 5. Concluding remarks are in the final section.

2 STOCHASTIC ASSET-LIABILITY MANAGEMENT MODEL

ALM is focused on modeling suitable sample paths for the assets and liabilities of a pension fund, bank, insurance company, or any other institution that dynamically manages and matches risks on both sides of a balance sheet (see e.g. Mulvey & Ziemba, 1998; Zenios & Ziemba, 2006; Hochreiter & Pflug, 2007; Haneveld et al., 2010; Mitra & Schwaiger, 2011; Righetto et al., 2016). In other words, the objective of ALM is to guarantee that liabilities are paid over a multi-period horizon by efficiently managing investment resources (e.g. Ziemba, 2003; Matos et al., 2014). Our study's model is designed for the pension fund environment. Thus, the fund's assets must be strategically managed so that the total value of all assets is greater than the fund's liabilities with high probability, while respecting all constraints, for instance, the fund's solvency (Bogentoft et al., 2001).

The stochastic programming community has discussed the ALM for a while. Early models were proposed by Bradley & Crane (1972), with commercial versions emerging thereafter for banks and insurance companies, respectively, with Cariño et al. (1994); Kusy & Ziemba (1986) and Kosmidou & Zopounidis (2002). Others expanded these proposals, for instance, Boender (1997) presented a large-scale model in which the asset allocation resulted from heuristic techniques. More recently, multistage stochastic programming approaches have become a trend, with examples in Consigli & Dempster (1998); Kouwenberg (2001); Moraes & Faria (2016). Kouwenberg (2001) and Høyland & Wallace (2001) discussed the challenge of generating representative scenarios; a key aspect in multistage stochastic programming. Stochastic programming ALM models related to non-neutral risk measures and based on the CVaR measure, or including jumps in asset prices, have already been presented in Rockafellar & Uryasev (2000); Bogentoft et al. (2001); Kilianová & Pflug (2009); Ferstl & Weissensteiner (2011); Josa-Fombellida & Rincón-Zapatero (2012).

Our stochastic ALM model is based on de Oliveira et al. (2017). Suppose that there is a set of securities denoted by $i = 1, \dots, N$. The ALM manager has to choose from these investment opportunities, allocating the available wealth in order to afford the liabilities denoted by l_t . These decisions occur repeatedly through different time periods, $t = 1, \dots, T$. The utility function maximizes the final expected wealth. Thus, the model takes the form of a stochastic and inter-temporal dynamic allocation problem, due to the randomness of the asset prices and the time-dependent nature of the investment and rebalancing decisions.

The investment strategy is designed with three sets of variables. The here-and-now decisions that are taken before the information is revealed, denoted by X_{its} . This is the position (shares) of asset i to hold in time period t and scenario s . The corrective actions, which are made after information revealing, are described by both B_{its} and V_{its} wait-and-see variables. They are, respectively, the position (shares) of i bought and sold during time t and scenario s .

The input data of the optimization model is determined by the continuous random variable ξ . This stochastic variable may be discretized by ξ_{it} that defines the price of asset i at time t . Additionally, it can take a finite number, $s \in 1, \dots, S$, of realizations denoted by P_{its} . As the asset returns are simulated by stochastic process, detailed in Section 4.1, the price realization P_{its} is defined as a conditional sampling of the random variable ξ_{it} . Therefore, the realizations for P_{its} depend on P_{it-1s} (Shapiro et al., 2009).

The scenario tree is comprised of scenarios ranging from the initial to the final period. A scenario k consists of a set of sequential realizations, $\{P_{i1s}, P_{i2s}, \dots, P_{iT_s}\} \forall s \in S$, of the random variable ξ_{it} . They must be adapted to the non-anticipative constraints that drive the information unfolding process. For instance, the scenario k is a random vector, generated by a ξ_{it} realization, which is described as $\lambda_k(\xi_{it}) = (P_{i1k}, \dots, P_{itk})$, $\forall i \in N$. Table 1 presents a summary of the model's notations.

The model is a multistage stochastic programming model with chance constraints, described below:

$$\max \quad \sum_{s=1}^S \sum_{i=1}^N \phi_{Ts} P_{iT_s} X_{iT_s} \quad (3)$$

s.t. :

$$Q = \sum_{i=1}^N P_{i0} X_{i0} \quad (4)$$

$$X_{its} = X_{i(t-1)s} + B_{its} - V_{its}, \quad \forall t \in T, \forall i \in N, \forall s \in S \quad (5)$$

$$\mathbb{P} \left(\sum_{i=1}^N \xi_{it} X_{it} \geq K(L_t - F_t) \right) \geq \alpha_t, \quad \forall t \in T, \quad (6)$$

Table 1 – Notation Summary.

Sets – Indices	
t	time index (stage) $t = 0, 1, \dots, T$
i	index of asset classes $i = 1, \dots, N$
s	index of scenarios $s = 1, \dots, S$
Decision variables	
X_{its}	Number of shares of assets i to hold during time t and scenario s
B_{its}	Number of shares of assets i to buy during time t and scenario s
X_{i0}	Number of shares of assets i to hold initially ($t = 0$)
V_{its}	Number of shares of assets i to sell during time t and scenario s
Random variables	
ξ_{it}	Random price of asset i during time t
Deterministic parameters	
Q	Initial wealth
α_t	Reliability level during time t
K	Legally required funding ratio
L_t	Present value of future liability $t = t + 1, \dots, T$
l_t	Liability to be paid during period t
F_t	Present value of future external contributions, $t = t + 1, \dots, T$
f_t	External contributions in each period t
M	Maximum amount of underfunding allowed
ρ	Discounting factor
π	Maximum weight of an asset in the portfolio
ϕ_{ts}	Probability of scenario s at time t
P_{its}	Price of asset i at time t and in scenario s
P_{i0}	Known initial ($t = 0$) price of asset i

$$\sum_{i=1}^N P_{its} V_{its} - \sum_{i=1}^N P_{its} B_{its} + f_t = l_t, \quad \forall t \in T, \forall s \in S \quad (7)$$

$$X_{its} P_{its} \leq \pi \sum_{i=1}^N X_{its} P_{its}, \quad \forall t \in T, \forall i \in N, \forall s \in S \quad (8)$$

$$X_{itk} = X_{itk'} \text{ if } \lambda_k(\xi_{it}) = \lambda_{k'}(\xi_{it}), \quad \forall t \in T, \forall i \in N, k, k' \in S \quad (9)$$

$$X_{its}, B_{its}, V_{its} \geq 0, \quad \forall t \in T, \forall i \in N, \forall s \in S \quad (10)$$

The model maximizes the expected terminal value of the fund (3). The objective function reflects the fund manager's goal to reach the largest possible gains in the last period, while respecting the risk and liability payment constraints. At time $t = 0$, the constraint (4) is deterministic, so that the initial wealth Q is distributed among the asset classes. As the price P_{i0} of each asset is

known and deterministic, we can rewrite X_{i0s} as X_{i0} , $\forall s$. The number B_{i0} of shares bought is equal to the number X_{i0} ($\forall i$) of shares kept at the end of the initial period. The linear equalities (5) are the share balance constraints. These specify that the number of shares X_{its} of asset i during time t and scenario s is equal to the number of shares $X_{i(t-1)s}$ maintained in the previous period, augmented by the number of purchased shares B_{its} minus those sold V_{its} at time t .

The chance constraints (6) enforce the funding ratio requirements, which represent the long-term relationship between assets and liabilities. The actual funding ratio β_{ts} of a fund during time t and scenario s is computed as:

$$\beta_{ts} = \frac{F_t + \sum_{i=1}^N P_{its} X_{its}}{L_t}, \quad t = 1, \dots, T, s = 1, \dots, S, \quad (11)$$

where $\sum_{i=1}^N P_{its} X_{its}$ is the current asset value of the pension fund in scenario s . L_t and F_t are, respectively, the present value of the future liabilities and contributions discounted by ρ :

$$L_t = \sum_{j=t}^T \frac{l_j}{(1+\rho)^{j-t}}, \quad F_t = \sum_{j=t}^T \frac{f_j}{(1+\rho)^{j-t}}.$$

A value of β less than 1 signals that the value of the assets may become insufficient to cover future liabilities, so the fund might run into solvency issues in the near future. The two parameters K and α_t define the asset-liability management policy's risk-aversion. The constraints (6) can be viewed as some sort of VaR constraints, ensuring that the value of the fund is at least equal to $K(L_t - F_t)$ during each period t with a probability at least equal to α_t . These maintain the fund's long-term solvency level.

The model cash flow balance is denoted by the equalities indicated by Eq. (7). The financial inputs are represented by the sales of assets and external contributions, while the financial outputs are payments of liabilities and purchases of assets. The constraint (8) stipulates that no asset can have a weight greater than an upper bound π . Multistage models have the so-called non-anticipativity condition. This determines that decisions are impacted only by the past, not by the future, i.e. two scenarios with a common history until the t th stage should result in the same decisions until this stage (Carøe & Schultz, 1999). Constraint (9) guarantees that scenarios with the same history present identical asset allocation. A maximum admissible underfunding value is defined through the parameter M . Constraint (10) defines the non-negativity restriction on the decision variables.

3 SCENARIO GENERATION METHODS IN ALM

In this section, we explain the simulation algorithms for generating scenario trees for the multistage decision problems used in our study: Monte Carlo sampling, Moment matching and Re-sampled average approximation. They gather the stochastic processes into a multistage scenario tree to model uncertainty. Even though the Monte Carlo with naive allocation strategy is not a scenario generation method, its explanation is still kept in this section.

3.1 ALM with classical Monte Carlo sampling

The traditional method to generate a scenario tree for ALM is through Monte Carlo sampling, with uniformly distributed pseudo-random numbers transformed appropriately into a target distribution (Ubeda & Allan, 1994; Dempster et al., 2011). This approach is an efficient way to represent multi-dimensional distributions (Zenios & Ziemba, 2006). In this study, we generate W_1, \dots, W_I random vectors from the standard normal distribution. As Homem-de-Mello & Bayraksan (2014a) note, in this case, the vectors W_1, \dots, W_I are mutually independent; a detail that characterizes this sampling method.

As Monte Carlo is based on the volume of a set distribution for the definition of probability measure, an obvious way to deal with this problem is to increase the number of nodes in the randomly sampled event tree. However, the stochastic program might become computationally intractable due to the tree's exponential growth rate. This hypothesis is supported, not only by the law of large numbers that guarantees the convergence to a correct value as the number of draws increase, but also by the central limit theorem that offers information about the error magnitude after a finite number of simulations. Hence, the convergence and error estimation of the outputs is directly linked with the number of draws. Indeed, one of the features of Monte Carlo is the form of the standard error, which, for a generic function f , can be defined by σ_f/\sqrt{n} , with n being the number of draws, and σ_f the sample standard deviation. The Monte Carlo sampling is also a good choice for integrals in high dimensions, because its convergence rate holds ($O(n^{-1/2})$) for any dimension. Glasserman (2003); Rubinstein & Kroese (2016) provide more specific features of Monte Carlo sampling.

According to Kouwenberg (2001), despite the intuitiveness of this approach, the mean and covariance matrix may not be correctly specified in most nodes of the tree, given that the states are randomly sampled. Thus, the optimizer might choose an investment strategy from erratic or misspecified parameters.

3.2 ALM with Moment matching sampling

The Moment matching approach aims to mitigate the impact of the inconsistencies in the specification, given that it is not possible to reach a full match with misspecified parameters. It also allows the decision maker to determine the output features based on the statistical distribution properties considered relevant.

Initially, as argued by Smith (1993), and proposed by Høyland & Wallace (2001), the Moment matching sampling matches the statistical properties to minimize the error between the sampled data of the tree and the first two moments of a theoretical distribution. Therefore, the scenario tree keeps some statistical properties reflecting the same characteristics as the theoretical distribution. Since there are a finite number of moments m on a continuous distribution, in our case two, Dupačová et al. (2000) argue that there always exists a discrete probability with these same moments and its support has at most $m + 2$ points (see Prékopa, 1995, chapter 5 for the proof).

Furthermore, the literature presents examples in which higher order moments are matched (Høyland et al., 2003). However, it may become very difficult to obtain a solution for non-linear constraints such as skewness and kurtosis. Kouwenberg (2001) and Zenios & Ziemba (2006) also adjust only the first two moments in their analysis. In this study, we construct an event tree that fits the mean and the covariance matrix of the underlying distribution.

In order to fit the first two moments, we use Cholesky decomposition in the same direction as in Høyland et al. (2003). The first step is to generate random vectors from the standard normal distribution, as described in Section 3.1. Then, the I random vectors are transformed to show a given covariance matrix by multiplying the vectors by a lower triangular matrix L of the covariance matrix Σ ,

$$W'_j = L W_j, \Sigma = L L', \quad j = 1, \dots, I, \quad (12)$$

where we can obtain L by applying Cholesky decomposition. In other words, as Kouwenberg (2001) states, we specify that the average of the disturbances should be zero, and they should have a covariance matrix equal to Σ . Therefore, we denote this matching in Eqs. (13) and (14):

$$\frac{1}{S} \sum_{s=1}^S W_{js} = 0 \quad \forall j \in 1, \dots, I, \quad (13)$$

$$\frac{1}{S-1} \sum_{s=1}^S W_{js} W_{is} = \Sigma_{ij} \quad \forall j, i \in 1, \dots, I. \quad (14)$$

This methodology allows for the generation of different sample paths, which matches the first two moments since the disturbance compose the asset price modeling (see Equations 16 and 17). As in Kouwenberg (2001); Dempster et al. (2011) and Löhndorf (2016), it is possible to argue that this sampling approach outperforms other methods, such as Monte Carlo sampling, Wasserstein distance sampling, or even Latin hypercube sampling. This methodology also enables us to capture unlikely scenarios if we consider higher order moments (Dupačová et al., 2000; Høyland & Wallace, 2001).

Unlike Monte Carlo sampling, the Moment matching does not necessarily converge when the number of scenarios is increased. It depends mainly on how much statistical properties are being matched and the distribution features, for example, the smoothness (Kaut & Wallace, 2007). Overspecification and underspecification are also an issue when dealing with the first two moments (Høyland & Wallace, 2001).

The works of Høyland & Wallace (2001) and Høyland et al. (2003) have been also adapted during the last few years. For instance, Gülpinar et al. (2004) discuss adaptations of Moment matching using sequential and overall optimization. Beraldi et al. (2010) propose a variant of Moment matching using parallel processing techniques, in which the scenario probabilities also become decision variables. Without being extensive, Mehrotra & Papp (2013) make the case for a more flexible optimization method in which some lower-order moments can be matched exactly, while higher-order moments only approximately.

3.3 ALM with the Resampled average approximation

The Resampled average approximation is a simulation of many scenario trees, with the average of the initial portfolio taken as the solution (de Oliveira et al., 2017). When we consider several trees, we account for a wide spectrum of variability inherent to the parameters in the optimization problem. This technique has some similarities to the Resampled efficient frontier method proposed by Michaud & Michaud (2008) to construct portfolios of risky securities. With a mathematical definition similar to that for the Resampled efficient frontier method for mean-variance optimal portfolios, the ALM Resampled average approximation optimality is the expected value in the solution space of the Monte Carlo ALM financial plan, as presented in Section 3.1. Thus, it may be seen as a reshuffle of the classical Monte Carlo. This methodology could also be adapted to the Moment matching, but the volatility of its results has already been regulated through the adjustment of the second moment. In other words, it is a needless further procedure to mitigate the risk, which is supposedly one of the main advantages of the Moment matching when compared to the classical Monte Carlo sampling.

This sampling technique is applied and described in detail by de Oliveira et al. (2017). Additionally, this approach can be viewed as a particular case of one of the algorithms discussed by Homem-de-Mello & Bayraksan (2014a), in which the stopping criteria is predefined by the number of runs. The ALM with the Resampled average approximation has four steps. First, we define the number of trees to solve for each parametrization (see e.g. Michaud & Michaud, 2008). The value of these outcomes must be sufficiently large to provide stable portfolio allocations, while being small enough to ensure that the approach does not become computationally prohibitive. After defining the number of instances, the second step is to generate the scenarios for each tree according to the ALM model. In Step 3, we solve each corresponding optimization problem to optimality. In Step 4, we evaluate the results based on the optimal solutions of the trees. The average of the initial allocation portfolio of all simulations is then the model solution. It is distinguished from SAA because of the number of scenario trees that are solved. In the SAA, a unique scenario tree is generated and solved. In this case, the expectation is applied on the decisions from the next stage. Unusually, the Resampled average approximation gives origin and solves an arbitrary number of unrelated scenario trees with the same topology, but for distinct scenarios. After that, the expectation is taken from these independent instances.

3.4 ALM with the Monte Carlo with naive allocation strategy

The Monte Carlo with naive allocation strategy is generally a naive policy that is very often reported in the portfolio allocation literature, in contrast to mean-variance portfolio optimization (Benartzi & Thaler, 2001; DeMiguel et al., 2009). Although this methodology is heuristic and does not reflect rational behavior, this analysis is justified due to the estimation error inherent in the sampling process. The Monte Carlo with naive allocation assesses whether an optimized portfolio performs better, despite the misspecification data when compared to irrational behavior. DeMiguel et al. (2009) assert that, in the sample-based mean-variance strategy, a window with around 3,000 months is necessary for a portfolio with 25 assets to outperform the 1/N benchmark.

In ALM, banks have already used this rule to make their portfolios in the 1960s (Cohen & Hammer, 1967). Furthermore, pension funds also adopted this policy in the 1980s (Harrison & Sharpe, 1983), and the USA Pension Benefit Guaranty Corporation had followed this rule. Although the Monte Carlo with naive allocation, or the $1/N$ portfolio, might provide good returns, they may not be able to meet the legal requirements and cash balance constraints. Additionally, Zenios & Ziemba (2007) show that, as the naive allocation is unable to incorporate new information, stochastic programming can outperform it for ALM. Fleten et al. (2002) also compared the Fixed-Mix strategy with the multistage stochastic linear programming, verifying the superiority of its model over the Fixed-Mix approach. We consider the $1/N$ naive policy version for our tests as the benchmark for other sampling methods. We define our naive allocation as an equally distributed portfolio, described in Eq. (15):

$$X_{i0} = \frac{Q}{N} \cdot \frac{1}{P_{i0}} \quad i = 1, \dots, N. \quad (15)$$

Therefore, we use this popular investment policy with the classical Monte Carlo sampling, defined in Section 3.1, in order to settle a comparative level to other methods simulated. Even with the investment policy defined, we have to guarantee that constraints (4)–(10) are respected in this strategy. Therefore, in the simulations executed at Section 4, we set the funding ratio to one. Thus, in the beginning of the simulation's time horizon, the pension fund has the all necessary wealth to afford the liabilities until the final time period.

4 SIMULATION

First we introduce, in Section 4.1, arbitrage-free continuous-time stochastic processes used to simulate prices at a discrete set of dates (Glasserman, 2003). We adopt the Geometric brownian motion model (GBM) and the Cox-Ingersoll-Ross model (CIR), which is a single-factor term structure model. These stochastic processes are adapted to the scenario tree topology. Then, in Section 4.2, the scenario tree structure is described.

4.1 Generating sample paths

We generate the scenario trees by asset prices realization sampling. The asset prices follow correlated stochastic differential equations (SDEs). We use the GBM for stock prices (Neftci, 1996; Duffie, 2001):

$$d\xi_{1t} = \mu\xi_{1t}dt + \sigma\xi_{1t}dW_{1t}. \quad (16)$$

Thus, there is a GBM for each stock. For the price of a fixed income asset, we use the Cox-Ingersoll-Ross term structure model (Cox et al., 1985):

$$d\xi_{2t} = \alpha(\mu - \xi_{2t})dt + \sqrt{\xi_{2t}}\sigma dW_{2t}, \quad (17)$$

where ξ_{2t} is the interest rate and (α, μ, σ) are model parameters. The drift function $\alpha(\mu - \xi_{2t})$ is linear and has a mean reverting property, i.e. the interest rate ξ_{2t} moves in the direction of

its mean μ at speed α . The diffusion function $\xi_{2t}\sigma^2$ is proportional to the interest rate ξ_{2t} and ensures that the interest rate is always positive.

The total of succeeding nodes for each scenario s at time t (defined by Table 2) are available to describe the conditional distribution of these random variables in a particular node at time $t - 1$. We define the disturbance W_{js} as the realization in node s for the j th element of the vector W . The model maximizes the expected wealth of an ALM problem applied to a pension fund, as defined generically in Eq. (1) respecting a set of restrictions.

We use only broad stock or bond indexes that cause arbitrage in cases of very poor approximation of the assets' underlying distribution on the stochastic programming tree (Kouwenberg & Zenios, 2006). We use just two stock broad indexes and a floating rate fixed income instrument with long-only positions, without any complex asset. We also present a large number of scenarios for each tree. Thus, in our study settings, the discussion of arbitrage is not critical.

The parameters used in GBM, Eq. (16) are estimated through historical time series data. Those used in CIR, Eq. (17), are estimated using maximum likelihood. These sampled paths are conditioned to non-anticipative constraints defined in Eq. (9), following the multistage stochastic framework integrated with the sampling algorithms. We simulate the classical Monte Carlo sampling, the Moment matching method, the Resampled average approximation and Monte Carlo with naive allocation, as explained in Section 3.

4.2 Scenario tree simulation

The simulation occurs prior to the optimization process; this methodology is known as the non-recursive method, (see Pflug, 2012). In order to simulate the trees and to study the outputs of the sampling methods, we define three classes of scenario trees that are denoted by small, medium and large according to their number of periods. They have 4, 6 and 8 periods respectively. In each class, we determine a different topology, which produces a distinct number of nodes, scenarios, variables and constraints (Table 2).

Table 2 – Classes and Topologies of scenario trees.

Name	Topology	Nodes	Scenarios	Variables	Constraints
Small	1-27-9-9	2,458	2,187	24,843	17,203
Medium	1-81-3-3-3-3	9,802	6,561	101,253	68,613
Large_A	1-16-3-3-3-3-3-3	17,489	11,664	180,707	122,424
Large_B	1-8-6-3-3-3-3-3	17,481	11,664	180,603	122,359

In the large class, we test the behavior of scenario generation and optimization for two different topologies that present the same number of scenarios (11, 664). The number of variables and constraints shown in Table 2 is defined by the deterministic linear equivalent problem, denoted through the Eqs. (3) – (10). We compose each scenario with a discrete sequence of conditional

distributions of stocks and fixed income assets, calibrated with historical data from January 2012 to November 2016.

We use data from the Brazilian capital market. Equation (16) defines the stock price models, which are calibrated with annualized daily return prices from the Bovespa index (the most liquid stocks in the country) and the Brazilian Small Cap BM&F Bovespa index (stocks with small capitalization). We also have a fixed income asset modeled with Eq. (17) and calibrated with data from the 1-month Brazilian LTN (similar to a T-Bill in the USA) as a proxy for the short-term interest rate. In Table 3, we show the parameters used in the model.

Table 3 – CIR and GBM parameters.

Asset	Return Annualized (μ)	Std. Annualized (σ)	Mean Revert. (α)
Fixed Income	0.11297	0.04358	0.14599
Bovespa index	0.13503	0.23486	–
Small Cap index	0.07426	0.17716	–

The covariance matrix Σ takes the variance and covariance among Fixed Income, Bovespa index, and Small Cap index as inputs for Eq. (12) in the Moment matching sampling. In Table 4, we present the covariance matrix Σ observed from the data.

Table 4 – Covariance Matrix for the experiments.

Asset	Fixed Income	Bovespa index	Small Cap index
Fixed Income	0.005339086	-0.001021373	-0.000973024
Bovespa index	-0.001021373	0.055221863	0.035719690
Small Cap index	-0.000973024	0.035719690	0.031501802

In the Monte Carlo with naive allocation approach, we take the initial wealth and allocate it uniformly between the financial assets, as defined in Eq. (4). This position is fixed until the last period. In each period, we calculate the financial value of the portfolio and discount its liabilities. Monte Carlo with naive portfolio is not rebalanced in any time period. Thus, the final expected value of wealth is the sum of the net financial value of the portfolio multiplied by its probability of occurrence, similar to Eq. (3). Even though the Monte Carlo with naive allocation is not a sampling method, we take it into account as an element of comparison for our analysis.

We performed 20 simulations for each sampling method, then estimated the average and standard deviation from both the objective function and the initial allocation for each methodology applied for each class. Unlike the Random sample and Moment matching, where each turn is comprised of only one scenario tree, we generated and optimized 300 unrelated scenario trees to compose each turn of Resample average approximation. This procedure was applied on all topologies shown in Table 2. This analysis consists of 12, 280 runs, and evaluated 141, 426, 000 different scenarios. We designed the scenario generation in C++ and Matlab using AMPL and the CPLEX 12.6.1 solver to model and solve the optimization problems. We run it in a 64-bit desktop with

Intel Core i7-4510U 2GHz CPU and with 8GB of RAM. In Table 5, we report each class and method's computational times (mean and standard deviation) in seconds to solve a single tree.

Table 5 – Times elapsed for scenario generation and optimization.

Name	Tree Generation Time				Model Optimization Time				Total Time			
	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.	Mean	Std. Dev.	Min.	Max.
Small												
Moment matching	0.4	0.06	0.24	0.48	4	0.44	4	5	4.551	0.46	4.24	5.48
Monte Carlo	0.28	0.04	0.24	0.43	4.7	0.65	4	6	4.98	0.65	4.26	6.29
Resampling	0.23	0.02	0.19	0.29	4.36	0.37	3.56	5	4.60	0.38	3.75	5.25
Medium												
Moment matching	1.56	0.12	1.44	1.85	4.65	0.58	4	6	6.21	0.60	5.46	7.7
Monte Carlo	1.92	0.41	1.44	2.62	18.10	11.33	10	59	20.02	11.30	11.48	60.45
Resampling	1.81	0.18	1.59	2.12	16.586	2.26	13.66	21.02	18.39	2.39	15.39	23.14
Large												
Moment matching_A	5.18	1.21	4.5	8.92	14.15	1.66	12	19	19.33	2.63	16.61	27.92
Moment matching_B	6.04	1.26	4.61	8.51	16.6	3.20	12	23	22.64	3.89	16.61	30.38
Monte Carlo_A	5.84	1.39	4.51	8.55	25.1	9.48	11	49	30.94	9.99	15.51	54.14
Monte Carlo_B	6.11	1.03	4.82	8.67	19.70	5.13	14	38	25.81	5.50	20.13	45.56
Resampling_A	5.71	0.64	4.91	6.94	24.74	1.99	20.76	28.14	30.45	2.52	25.67	35.08
Resampling_B	6.35	0.68	5.22	7.66	19.02	3.81	4.10	23.32	25.37	3.91	11.28	30.72

Based on Table 5, we notice that, on average, the computational time to solve a single tree is between 4 and 61 seconds. For the Resampled average approximation, the computational experiment can be taken from 7.5 to 51 hours to be completed: 6, 000 trees must be computed for each instance. For the other three methods, the average time has to be multiplied by 20, which is the number of simulations to finish each experiment. The Monte Carlo with naive allocation strategy is not considered on Table 5 because it does not demand much computational effort. Next, we present and describe the results from the simulations.

5 RESULTS

Our results focus on the stochastic programming in-sample stability (Kaut & Wallace, 2007). The in-sample stability is evaluated by generating different scenarios for each tree and comparing how stable the problem's objective function and solution are. Thus, we analyze the stability of the sampling methods from two different perspectives: objective function and initial portfolio allocation. Dempster et al. (2011) argue that the initial portfolio allocation criterion is not often used in the literature, because of the potential for a flat plateau objective function. However, as our scenarios might have sampling error, we conducted the analyses from both perspectives. We focus on the expectation and volatility of both the objective function and initial portfolio allocation. The results of the objective function and initial portfolio allocation are presented in Tables 6 and 7, respectively.

In Table 6, we observe that objective function's standard deviation, of the Resampled average approximation, is the smallest for all classes and it is followed by the Moment matching. The Monte Carlo with naive approach and the Monte Carlo sampling present much larger standard deviations.

Table 6 – Objective function statistical outputs.

Name	Mean	Std. Dev.	Min.	Max.
Small				
Resampling	714,114.47	2,805.67	708,800.98	720,443.70
Moment matching	629,656	7,341.04	619,961	642,192
Naive allocation	814,619.86	30,153.65	762,498.94	864,254.46
Monte Carlo	712,815.40	22,927.45	672,949	756,231
Medium				
Resampling	1,065,780.19	3,850.40	1,056,652.26	1,073,008.54
Moment matching	711,529.65	8,698.65	695,761	727,839
Naive allocation	1,136,827.87	26,529.07	1,072,786.30	1,184,755.13
Monte Carlo	1,074,340.85	20,485.86	1,028,342	1,106,371
Large				
Resampling_A	1,584,145.66	18,498.18	1,554,806.82	1,633,666.16
Resampling_B	1,549,304.86	13,994.94	1,526,511.44	1,572,683.32
Moment matching_A	828,987	21,964.05	795,107	873,140
Moment matching_B	837,172.05	25,982.62	798,804	881,010
Naive allocation_A	1,564,422.42	68,972.70	1,469,548.25	1,736,670.72
Naive allocation_B	1,625,050.78	117,436.33	1,487,900.62	1,902,890.73
Monte Carlo_A	1,586,488.10	137,887.48	1,391,047	1,923,511
Monte Carlo_B	1,531,394.65	128,454.55	1,267,510	1,772,048

Table 7 – Initial allocation of decision variables (%).

Name	Fixed Income		Bovespa Index		Small Cap index	
	Mean (%)	Std. Dev.	Mean (%)	Std. Dev.	Mean (%)	Std. Dev.
Small						
Moment matching	44	0.28	56	0.28	0	0
Monte Carlo	77	0.08	20	0.07	3	0.10
Resampling	79.31	2.58	18.23	1.57	2.46	1.96
Medium						
Moment matching	37	0.21	63	0.21	0	0
Monte Carlo	77.76	0.01	22.00	0.01	0.24	0.00
Resampling	77.71	0.53	21.91	0.42	0.36	0.26
Large						
Moment matching_A	51.40	0.32	48.60	0.32	0	0
Moment matching_B	48.57	0.30	51.43	0.30	0	0
Monte Carlo_A	74.50	0.16	20.54	0.08	4.96	0.13
Monte Carlo_B	74.36	0.19	19.49	0.14	6.15	0.12
Resampling_A	75.13	2.30	19.60	1.48	5.27	2.17
Resampling_B	72.8	2.66	18.27	1.33	8.93	1.92

Furthermore, we notice that the average value of the objective function in the Moment matching is 11.66% and 12.19% lower compared, respectively, to the Monte Carlo sampling and to the Resampled average approximation in the small trees. This difference increases in large trees. In the medium class, it is 33.77% and 33.23% lower. For the large class, it becomes 47.74% and 47.66% for the instance Large_A and 45.33% and 45.96% for Large_B. We believe that this happens due to the adjustment in the tree. When the second moment is adjusted, volatility might be reduced and, thereby, the extreme scenarios (good or bad) are probably taken out of the sampling. Clearly, the Resampled average approximation and the Moment matching dominate the other two methods in terms of the stability of the objective function.

In relation to the initial asset allocation (Table 7), the investment in fixed income is quite similar for the Monte Carlo sampling, the Moment matching and Resampled average approximation. The main difference is that the Moment matching never allocates capital in the Small Cap index. The volatility of the Monte Carlo sampling and the Moment matching are also similar. The Resampled average approximation and Monte Carlo sampling present similar allocations, but very different allocation volatilities. However, given that the Resampled average approximation takes expectations from the Random sampling, the volatility of the allocation is close to zero. This is consistent with the methodology, as the Resampled average approximation mitigates volatility using the expected value from many trees, making the initial allocation smoother. Notice that, for the Monte Carlo with naive approach, by design, the initial allocation is constant among the assets, so we omit the results from Table 7.

In terms of the stability of the objective function, the Moment matching and Resampled average approximation dominate both Monte Carlo sampling and with naive allocation strategy. The stability of the initial allocation is similar for the Monte Carlo sampling and the Moment matching. Furthermore, the initial allocation's low volatility, presented by the Resampled average approximation, is a result of taking an average of averages. Considering not only the mixed results obtained on the initial asset allocation stability but also the argument of Dempster et al. (2011), that a flat plateau can make many different allocations, resulting in very close values of objective functions, we can focus on the objective function stability results. Then, clearly, the Moment matching and Resampled average approximation present more stable solutions.

6 CONCLUSIONS

In this study, we provided an empirical discussion of the differences among methods to generate scenario trees for stochastic programming in ALM. We examined Monte Carlo sampling, Moment matching, the Resampled average approximation and Monte Carlo with naive allocation strategy as the benchmarks.

Our empirical analysis approached financial scenario trees with three assets. We defined four distinct topologies, which were assigned in three classes (small, medium and large). We ran the simulations so that the model would be robust to represent a realistic environment, but we regarded the technological limitations. The number of assets is a key point for the size of financial

scenario trees. It should be arranged so that it is smaller than or equal to the number of scenario tree branches in order to guarantee the absence of arbitrage conditions (Geyer et al., 2010). Therefore, the model becomes computationally intractable very quickly when the number of assets rises, since they define a lower bound to number of branches, which can be used to calculate the exponential growth of scenario tree (Gülpinar et al., 2004).

The simulations are assessed by in-sample point of view. Thus, the best approximation is the one that minimizes the error between the “true” objective value and the optimal scenario generation method objective value (Kaut & Wallace, 2007). Considering the obtained results of the objective function, we can conclude that the Moment matching and the Resampled average approximation are more efficient in terms of in-sample stability, when compared to Monte Carlo sampling and Monte Carlo with naive allocation. For the initial portfolio allocation, the stability of the Moment matching and Monte Carlo sampling are similar and dominated by the Resampled average approximation. However, the low volatility of the Resampled average approximation is a result of taking an average of averages. Taking all the results into account, the Moment matching and the Resampled average approximation are more appropriate for ALM scenario generation.

There are other opportunities for subsequent studies. Scenario reduction and parallel implementation are techniques developed to deal with the curse of dimensionality in stochastic programming problems (Beraldi et al., 2010; Dupačová et al., 2003; Heitsch & Römisch, 2003). For instance, advances in the definitions of bounds (Henrion et al., 2009), different metrics (Heitsch & Römisch, 2007) and clustering (Beraldi & Bruni, 2014) have been proposed. The integration of these techniques with different scenario generation methods might present promising results.

ACKNOWLEDGEMENTS

This work was funded by the following Brazilian Research Agencies: CAPES and FAPERGS.

REFERENCES

- [1] BENARTZI S & THALER RH. 2001. Naive diversification strategies in defined contribution saving plans. *The American Economic Review*, **91**(1): 79–98.
- [2] BERALDI P & BRUNI ME. 2014. A clustering approach for scenario tree reduction: an application to a stochastic programming portfolio optimization problem. *TOP*, **22**(3): 934–949.
- [3] BERALDI P, DE SIMONE F & VIOLI A. 2010. Generating scenario trees: a parallel integrated simulation–optimization approach. *Journal of Computational and Applied Mathematics*, **233**(9): 2322–2331.
- [4] BIRGE JR & LOUVEAUX F. 2011. *Introduction to Stochastic Programming*. Springer Science+Business Media, New York, 2 edition.
- [5] BOENDER GCE. 1997. A hybrid simulation/optimisation scenario model for asset/liability management. *European Journal of Operational Research*, **99**(1): 126 – 135. ISSN 0377-2217.
- [6] BOGENTOFT E, ROMEIJN HE & URYASEV S. 2001. Asset/liability management for pension funds using cvar constraints. *The Journal of Risk Finance*, **3**: 57–71.

- [7] BRADLEY SP & CRANE DB. 1972. A dynamic model for bond portfolio management. *Management Science*, **19**: 139–151.
- [8] CARIÑO DR, KENT T, MEYERS DH, STACY C, SYLVANUS M, TURNER AL, WATANABE K & ZIEMBA WT. 1994. The Russel-Yasuda Kasai model: an asset/liability model for japanese insurance company using multistage stochastic programming. *Interfaces*, **24**(1): 29–49.
- [9] CARØE CC & SCHULTZ R. 1999. Dual decomposition in stochastic integer programming. *Operations Research Letters*, **24**(1): 37–45.
- [10] COHEN KJ & HAMMER FS. 1967. Linear programming and optimal bank asset management decisions. *The Journal of Finance*, **22**(2): 147–165.
- [11] CONSIGLI G & DEMPSTER MAH. 1998. Dynamic stochastic programming for asset – liability management. *Annals of Operations Research*, **81**: 131–161.
- [12] COX JC, INGERSOLL JE & ROSS SA. 1985. A theory of the term structure of interest rates. *Econometrica*, **53**(2): 385–408.
- [13] DE OLIVEIRA AD, FILOMENA TP, PERLIN MS, LEJEUNE M & DE MACEDO GR. 2017. A multistage stochastic programming asset-liability management model: an application to the brazilian pension fund industry. *Optimization and Engineering*, **18**(2): 349–368.
- [14] DEMIGUEL V, GARLAPPI L & UPPAL R. 2009. Optimal versus naive diversification: how inefficient is the $1/n$ portfolio strategy? *Review of Financial Studies*, **22**(5): 1915–1953.
- [15] DEMPSTER MAH, MEDOVA EA & YONG YS. 2011. Comparison of sampling methods for dynamic stochastic programming. In *Stochastic Optimization Methods in Finance and Energy*, chapter 16, pages 389–425. Springer Science+Business Media, New York.
- [16] DUFFIE D. 2001. *Dynamic Asset Pricing Theory*. Princeton University Press, Princeton.
- [17] DUPAČOVÁ J, CONSIGLI G & WALLACE SW. 2000. Scenarios for multistage stochastic programs. *Annals of Operations Research*, **100**(1): 25–53.
- [18] DUPAČOVÁ J, GRÖWE-KUSKA N & RÖMISCH W. 2003. Scenario reduction in stochastic programming. *Mathematical Programming*, **95**(3): 493–511.
- [19] ESCUDERO LF & KAMESAM PV. 1995. On solving stochastic production planning problems via scenario modelling. *TOP*, **3**(1): 69–95.
- [20] FERSTL R & WEISSENSTEINER A. 2011. Asset-liability management under time-varying investment opportunities. *Journal of Banking and Finance*, **35**(1): 47–62.
- [21] FLETEN SE, HØYLAND K & WALLACE SW. 2002. The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, **140**(1): 37–49.
- [22] GEYER A, HANKE M & WEISSENSTEINER A. 2010. No-arbitrage conditions, scenario trees, and multi-asset financial optimization. *European Journal of Operational Research*, **206**(3): 609–613.
- [23] GLASSERMAN P. 2003. *Monte Carlo methods in Financial Engineering*, volume 53. Springer-Verlag New York, New York, 1 edition.
- [24] GÜLPINAR N, RUSTEM B & SETTERGREN R. 2004. Simulation and optimization approaches to scenario tree generation. *Journal of Economic Dynamics and Control*, **28**(7): 1291–1315.

- [25] HANEVELD WKK, STREUTKER MH & VAN DER VLERK MH. 2010. An ALM model for pension funds using integrated chance constraints. *Annals of Operations Research*, **177**(1): 47–62. ISSN 0254-5330.
- [26] HARRISON JM & SHARPE WF. 1983. Optimal funding and asset allocation rules for defined-benefit pension plans. In *Financial Aspects of the United States Pension System*, pages 91–106. University of Chicago Press.
- [27] HEITSCH H & RÖMISCH W. 2003. Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, **24**(2-3): 187–206.
- [28] HEITSCH H & ROMISCH W. 2005. Generation of multivariate scenario trees to model stochasticity in power management. In *IEEE St. Petersburg Power Tech*.
- [29] HEITSCH H & RÖMISCH W. 2007. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, **35**(6): 731–738.
- [30] HENRION R, KÜCHLER C & RÖMISCH W. 2009. Scenario reduction in stochastic programming with respect to discrepancy distances. *Computational Optimization and Applications*, **43**(1): 67–93.
- [31] HOCHREITER R & PFLUG G. 2007. Financial scenario generation for stochastic multi-stage decision processes as facility location problems. *Annals of Operations Research*, **152**(1): 257–272.
- [32] HOMEM-DE-MELLO T & BAYRAKSAN G. 2014a. Monte carlo sampling-based methods for stochastic optimization. *Surveys in Operations Research and Management Science*, **19**(1): 56–85.
- [33] HOMEM-DE-MELLO T & BAYRAKSAN G. 2014b. Stochastic constraints and variance reduction techniques. In M.C. Fu, editor, *Handbook of Simulation Optimization*, volume 216, pages 245–276. International Series in Operations Research and Management Science – Springer.
- [34] HØYLAND K & WALLACE SW. 2001. Generating scenario trees for multistage decision problems. *Management Science*, **47**(2): 295–307.
- [35] HØYLAND K, KAUT M & WALLACE SW. 2003. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, **24** (2): 169–185.
- [36] JOSA-FOMBELLIDA R & RINCÓN-ZAPATERO JP. 2012. Stochastic pension funding when the benefit and the risky asset follow jump diffusion processes. *European Journal of Operational Research*, **220**(2): 404–413. ISSN 0377-2217.
- [37] KAUT M & WALLACE SW. 2007. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, **3**(2): 257–271.
- [38] KAUT M & WALLACE SW. 2011. Shape-based scenario generation using copulas. *Computational Management Science*, **8**(1): 181–199.
- [39] KILIANOVÁ S & PFLUG GC. 2009. Optimal pension fund management under multi-period risk minimization. *Annals of Operations Research*, **166**(1): 261–270. ISSN 0254-5330.
- [40] KOSMIDOU K & ZOPOUNIDIS C. 2002. An optimization scenario methodology for bank asset liability management. *Operational Research*, **2**(2): 279–287.
- [41] KOUWENBERG R. 2001. Scenario generation and stochastic programming models for asset liability management. *European Journal of Operational Research*, **134**(2): 279–292.
- [42] KOUWENBERG R & ZENIOS SA. 2006. Stochastic programming models for asset liability management. In *Handbook of asset and liability management*, volume 1, chapter 6, pages 253–303. Elsevier.

- [43] KUSY MI & ZIEMBA WT. 1986. A bank asset and liability management model. *Operations Research*, **34**: 356–376.
- [44] LÖHNDORF N. 2016. An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research*, **255**(1): 121–132.
- [45] MAK WK, MORTON DP & WOOD RK. 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, **24**(1): 47–56.
- [46] MATOS P, PADILHA G & BENEGAS M. 2014. On the management efficiency of brazilian stock mutual funds. *Operational Research*, pages 1–35.
- [47] MCKAY MD, BECKMAN RJ & CONOVER WJ. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**(2): 239–245.
- [48] MEHROTRA S & PAPP D. 2013. Generating moment matching scenarios using optimization techniques. *SIAM Journal on Optimization*, **23**(2): 963–999.
- [49] MICHAUD R & MICHAUD R. 2008. *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. Oxford University Press, 2 edition.
- [50] MITRA G & SCHWAIGER K, EDITORS. 2011. *Asset and Liability Management Handbook*. Palgrave Macmillan, 1 edition.
- [51] MORAES LAM & FARIA LFT. 2016. A stochastic programming approach to liquefied natural gas planning. *Pesquisa Operacional*, **36**(1): 151–165.
- [52] MULVEY JM & ZIEMBA WT. 1998. Asset and liability management systems for long-term investors: Discussion of the issues. In *Worldwide Asset and Liability Modeling*, pages 3–40. Cambridge University Press.
- [53] NEFTCI SN. 1996. *An Introduction to the Mathematics of Financial Derivatives*. Academic Press, United States, 1 edition.
- [54] PFLUG GC. 2001. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, **89**(2): 251–271.
- [55] PFLUG GC. 2012. *Optimization of Stochastic Models: The Interface between Simulation and Optimization*, volume 373. Springer Science & Business Media.
- [56] PRÉKOPA A. 1995. Moment problems. In *Stochastic Programming*, chapter 5, pages 125–178. Springer Netherlands, Budapest.
- [57] RIGHETTO GM, MORABITO R & ALEM D. 2016. A robust optimization approach for cash flow management in stationery companies. *Computers & Industrial Engineering*, **99**: 137–152.
- [58] ROCKAFELLAR RT & URYASEV S. 2000. Optimization of conditional value-at-risk. *Journal of Risk*, **2**: 21–41.
- [59] ROMISCH W. 2003. Stability of stochastic programming problems. In A. Ruszczyński and A. Shapiro, editors, *Stochastic Programming, Volume 10 of Handbooks in Operations Research and Management Science*, chapter 8, pages 483–554. Elsevier, Amsterdam.
- [60] RUBINSTEIN RY & KROESE DP. 2016. *Simulation and the Monte Carlo method*. John Wiley & Sons.

- [61] SHAPIRO A, DENTCHEVA D & RUSZCZYŃSKI A. 2009. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 1 edition.
- [62] SMITH JE. 1993. Moment methods for decision analysis. *Management Science*, **39**(3): 340–358.
- [63] UBEDA JR & ALLAN RN. 1994. Stochastic simulation and monte carlo methods applied to the assessment of hydro-thermal generating system operation. *TOP*, **2**(1): 1–23.
- [64] ZENIOS SA & ZIEMBA WT. 2006. *Handbook of Asset and Liability Management – Volume 1: Theory and Methodology*. Elsevier, United Kingdom, 1 edition.
- [65] ZENIOS SA & ZIEMBA WT. 2007. *Handbook of Asset and Liability Management – Volume 2: Applications and Case Studies*. Elsevier, United Kingdom, 1 edition.
- [66] ZIEMBA WT. 2003. *The Stochastic Programming Approach to Asset, Liability and Wealth Management*. AIMR Publisher, Vancouver, Canada and Londres, England.